REMARKS/ARGUMENTS

By this Amendment, Claims 1-21 are canceled and Claims 22-39 are added. Thus, Claims 22-39 are pending in this application.

Applicants have assumed that the Preliminary Amendment filed on July 23, 2001 corrected the duplicative claim numbering (Claims 10-18 following Claims 1-12 in the originally-filed application) resulting in Claims 1-21 known hereafter as "Previous Set of Claims in Application." Applicants have cancelled Claims 1-18 in favor of new corresponding Claims 22-39. Claims 19-21 were cancelled and have no new corresponding claims.

Previous Set of Claims in Application	Currently-Pending Set of Claims in Application
1	22
2	23
3	24
4	25
5	26
6	27
7	28
8	29
9	30
10	31
11	32

Previous Set of Claims in Application	Currently-Pending Set of Claims in Application
12	33
13	34
14	35
15	36
16	37
17	38
18	39

The Applicants' note that the Examiner also sets forth that the lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors and that Applicants' cooperation is requested in correcting any errors of which applicant may become aware in the specification.

The Examiner further sets forth that Claim 14 (now Claim 35) is objected to because the Examiner believes that line 1, "examining" should be "overwriting" for the claims to make sense. Appropriate correction is requested by the Examiner. Accordingly, Claim 35 has been amended in accordance with the Examiner's objection.

According to the Examiner, Claims 1-3, 5, 7, 8, 10, 12-15 and 17 (corresponding to Claims 22-24, 26, 28, 29, 31, 33-36 and 38) are rejected under 35 U.S.C. 102(e) as being anticipated by Houldsworth (U.S. 6,314,436).

Regarding Claim 22, the Examiner believes that Houldsworth discloses all the claimed subject matter (the Examiner directs the Applicants' attention to the abstract, Figures 1-8, column 2, line 12- column 3, line 60). The claimed "reading a start pointer of

one of the entries" is met when Houldsworth shows checking whether the start pointer is set to null and starting with an item that carries a pointer according to the Examiner. Additionally, the Examiner believes that the claimed "examining the entries...is found" and "accessing the particular entity" are met when Houldsworth shows determining the items referenced by the item under consideration, selecting next items and determining whether this item is white (the Examiner directs the Applicants' attention to column 6, lines 20-60). The Examiner further sets forth the claimed "overwriting the start pointer so as to point to the particular entry" is met by the fact that the start pointer is updated once all items referenced by an item under consideration have been dealt with (the Examiner directs the Applicants' attention to Figures 2-4, 7 and 8).

The Examiner sets forth that Claim 28 is essentially Claims 22 and 23 combined, and is thus rejected by the Examiner for the same reasons stated in Claims 22 and 23 above.

According to the Examiner, Claim 33 merely differs from Claim 22 by reciting a "reference" instead of a "pointer" and the pointer in Houldsworth is clearly a reference. Therefore, Claim 33 is rejected by the Examiner for the same reasons stated in Claim 22 above.

Regarding Claims 23 and 34, the Examiner sets forth that the claimed "terminating examination...have been examined" merely reads on the fact that the method of Houldsworth terminates when the stack is empty (the Examiner directs the Applicants' attention to Figure 6).

Regarding Claims 24, 29 and 36, the Examiner sets forth that Houldsworth discloses reading the entry (the Examiner directs the Applicants' attention to Figures 2-4).

Regarding Claims 26, 31 and 38, the Examiner sets forth that clearly the method of Houldsworth does not require a mutex for overwriting the start pointer since the Examiner believes that Houldsworth does not mention any locking mechanism.

According to the Examiner, Claim 35 merely reads on the fact that once the items at the bottom level of the list of items have been determined, the operation terminates (the Examiner directs the Applicants' attention to Figures 2-4).

The Examiner sets forth that Claims 25, 27, 30, 32, 37 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Houldsworth (U.S. 6,314,436), in view of Fresko et al. (U.S. 6,349,312).

Regarding Claims 25, 30 and 37, the Examiner sets forth that Houldsworth discloses writing to the entry when Houldsworth shows that the method changes the color of an item (the Examiner directs the Applicants' attention to Figures 2-4). Although the Examiner believes that Houldsworth does not specifically show imposing a mutex, according to the Examiner it is well known in the art as shown by Fresko to use a mutual exclusive operation on multiple requests (the Examiner directs the Applicants' attention to column 6, lines 50-66). Therefore, the Examiner believes that it would have been obvious to one of ordinary skill in the art to include the claimed features while implementing the method of Houldsworth in order to prevent concurrent updates to an entry.

Applicants' Discussion of the Prior Art

Regarding Claims 27, 32 and 39, the Examiner sets forth that although Houldsworth does not specifically show a multi-threaded computer system, the Examiner believes that it is well known in the art as shown by Fresko to enhance a single-threaded system to a multi-threaded system (the Examiner directs the Applicants' attention to column 6, lines 39-49). Therefore, the Examiner believes that it would have been obvious to one of ordinary skill in the art to include the claimed features while implementing the method of Houldsworth in order to allow sharing of memory as taught by Fresko.

Houldsworth teaches a method for tracing paths through stored data structures defined by data objects coupled to each other by way of identifying pointers. When the addressing of the contents of a page of memory has been completed in the Houldsworth system, additional storage means may hold a pair of pointers that point to the starting locations of the first and last data objects that were accessed as the path through the data structures was traced [3:30-33].

Houldsworth also teaches providing a fixed-size mark store data structure 54 (see Fig. 5) for processing objects in a page, and a fixed-size page list data structure 56 for storing a reference to each page in the heap 50 (5:20-26). Additionally, a fixed-size table 58 having entries for all of the pages p.0-p.5 of the heap 50 is provided. Each entry in the table 58 includes the two pointer fields required for indicating the starting locations of the first and last data objects that were accessed (5:32-36), as previously described.

During the marking procedure taught by Houldsworth, the pointers of the table 58 are determined according to the starting locations of the first and last-accessed objects in

the page. No determinations other than the determinations of which object is accessed first and which object is accessed last are taught for determining the pointers. In particular, no statistical determinations are taught for the process of determining the pointers, since the pointers will always point to the first and last located data objects regardless of any statistical probabilities.

Fresko is directed to garbage collection. In particular, Fresko teaches a method and apparatus for performing pre-allocation of memory to avoid triggering of unwanted garbage collection operations. As set forth in Fresko, a contiguous memory space is pre-allocated and used to perform the actual memory allocation operations. Because the memory allocation operations are performed using memory space that has already been allocated, it is insured that the allocation operations will not trigger a garbage collection operation. Fresko is not directed to any conditional overwriting of pointers pointing to a list of data items in order to more efficiently search the list. Rather, Fresko is cited by the Examiner as teaching features such as the steps of imposing a mutex and enhancing a single-threaded system to a multi-threaded system which the Examiner believes are missing in Houldsworth.

Applicants' System

In contrast, the Applicants' invention is a system for accessing linked data structures wherein entries in a list of entries are provided with pointers for pointing to other entries in the list, the pointers forming a closed loop. Thus, access of the Applicants' list begins at

a start location indicated by a start pointer and proceeds from one list entry to another until the required data item is found. If a future access to the list were to begin at the same start location and follow the links to the new required data item, time would be wasted if the new required item is the same as the last-accessed data item during the previous access. Conversely, if the future access to the list were to begin with the last-accessed item, time could be lost if the new required data item is not the same as the last-accessed item in the previous access.

The Applicants have determined that in some applications it is possible to determine whether there is an above average probability that the last-accessed item will be the item that is being searched for in a future access. Furthermore, the Applicants have discovered that this statistical determination can be used to prevent unnecessary searching through a list. Therefore, the Applicants' inventive contribution is determining that list accesses can be made more efficient by making a statistical determination whether the last-accessed entry is more likely than not to be the next required item and overwriting the start pointer accordingly.

Thus, the Applicants' start pointer for a future access is overwritten with the address of the last-accessed data item if there is a statistical probability (with a magnitude greater than fifty percent) that the last-accessed data item will be accessed first during the future access.

With reference to the published application US2002/0165848 A1, the Applicant sets forth:

> In many applications, there is an above average probability that the particular entry which has been found will be the entry which is required the next time the list is accessed. In this case, with known implementation of the method, there is therefore an above average probability that it has been necessary to examine all the entries in the list before the particular entry is found, then on the next access it will also be necessary to examine all the entries in the list before the particular entry is found. (See Paragraph [1227])
>
> and
>
> In any environment where the list is unordered but there is an above average probability that the last list item found in a search of the list will also be asked for the next time the list is searched, then by changing the list_head as described above, we reduce a number of those visited in the search, and hence the search time. (See Paragraph [1270])

Thus, the Applicants teach: (1) determining whether there is such an above average probability that the last-accessed data item will be required during a future access, and (2) changing the start pointer accordingly.

Applicants' New Claims

Therefore, the Applicants' new Claim 22 sets forth a method of accessing an entry in a list of entries in a computer system, each entry in the list of entries including a next entry pointer that points to another entry in the list such that the next entry pointers together form a closed loop, the method including the step of statistically determining an access probability that a future access of the list will require a last-accessed entry in the list in order to provide a statistically determined entry. A further determination is made whether the access probability has a predetermined probability magnitude. A start pointer

pointing to one of the entries in the list is read and the entries in the list are examined in turn beginning with the entry pointed to by the start pointer and continuing until the statistically determined entry is found. The statistically determined entry is accessed and the start pointer is overwritten in accordance with the further determining in order to provide an overwritten pointer so as to point to the statistically determined entry when the access probability has the predetermined probability magnitude. The overwritten pointer is followed to the statistically determined entry during the future access to the list so as to make the statistically determined entry the first entry accessed during the future access to the list.

Applicants' new Claim 28 sets forth a method of accessing an entry in a list of entries in a computer system, each of the entries including a next entry pointer that points to another entry in the list such that the next entry pointers together form a closed loop, the method including the steps of statistically determining an access probability that a future access of the list will require a last-accessed entry in the list in order to provide a statistically determined entry and further determining whether the access probability has a predetermined probability magnitude. A start pointer pointing to one of the entries in the list is read and the entries in the list are examined in turn beginning with the entry pointed to by the start pointer and continuing until the statistically determined entry is found. If the statistically determined entry is found the following steps are performed: accessing the statistically determined entry and overwriting the start pointer in accordance with the further determining in order to provide an overwritten pointer so as to point to the statistically

determined entry when the access probability has the predetermined probability magnitude. If the statistically determined entry is not found and all of the entries in the list have been examined then the following step is performed: terminating examination of the entries.

Applicants' new Claim 33 sets forth a method of accessing an entry in a list of entries in a computer system, each of the entries including a next entry pointer that refers to another entry in the list such that the next entry pointers together form a closed loop, the method including the step of statistically determining an access probability that a future access of the list will require a last-accessed entry in the list in order to provide a statistically determined entry. Further determining whether the access probability has a predetermined probability magnitude is performed. A start pointer pointing to one of the entries in the list is read. The entries in the list are examined in turn beginning with the entry pointed to by the start pointer and continuing until the statistically determined entry is found and the statistically determined entry is accessed. The start pointer is overwritten in accordance with the further determining in order to provide an overwritten pointer so as to point to the statistically determined entry when the access probability has the predetermined probability magnitude. The overwritten pointer is followed to the statistically determined entry during the future access to the list so as to make the statistically determined entry the first entry accessed during the future access to the list.

Applicants' Patentability Arguments

Houldsworth does not teach any of the required features of the Applicants' new Claims 22, 28 and 33 directed to conditionally overwriting a start pointer in order to permit more efficient accesses of a list. These features, for which Houldsworth provides no teachings whatsoever, include: statistically determining an access probability that a future access of the list will require a last-accessed entry, further determining whether the access probability has a predetermined probability magnitude, and overwriting a start pointer in accordance with the further determination whether the access probability has the predetermined probability magnitude as set forth in.

Houldsworth does not teach or suggest any determination of any probability or any overwriting in accordance with a determination of a predetermined probability magnitude whatsoever. Houldsworth teaches only providing pointers to the starting address of the first and last-accessed data items regardless of any other determinations, including any probabilities. Therefore, Houldsworth does not teach or suggest the claimed invention of Claims 22, 28 and 33.

Fresko lacks the same required features of the Applicants' new Claims 22, 28 and 33 as set forth with respect to Houldsworth. Thus, Fresko does not teach or suggest the Applicants' new Claims 22, 28 and 33. Furthermore, any combination of Houldsworth and Fresko would also lack these required features. Additionally, Claims 23-27 depend from Claim 22, Claims 29-32 depends from Claim 28 and Claims 34-42 depend from Claim 33.

Therefore, for the reasons described above, these claims are also patentable over the art of record.

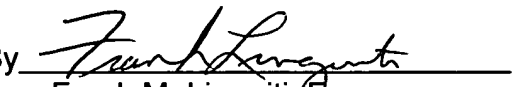
For at least the reasons set forth above, it is respectfully submitted that the above-identified application is in condition for allowance. Favorable reconsideration and prompt allowance of the claims are respectfully requested.

Should the Examiner believe that anything further is desirable in order to place the application in even better condition for allowance, the Examiner is invited to contact Applicants' undersigned attorney at the telephone number listed below.

Respectfully submitted,

CAESAR, RIVISE, BERNSTEIN,
COHEN & POKOTILOW, LTD.

August 6, 2004

By 

Frank M. Linguiti, Esq.
Registration No. 32,424
Customer No. 03000
(215) 567-2010
Attorneys for Applicants

Please charge or credit our Account No. 03-0075 as necessary to effect entry and/or ensure consideration of this submission.